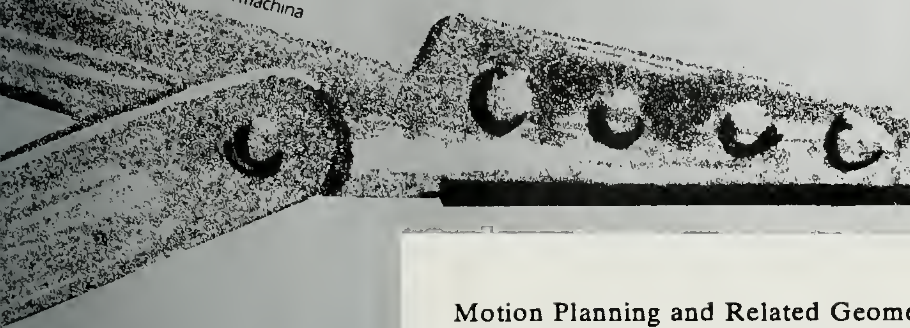
Robotics Research Technical Report

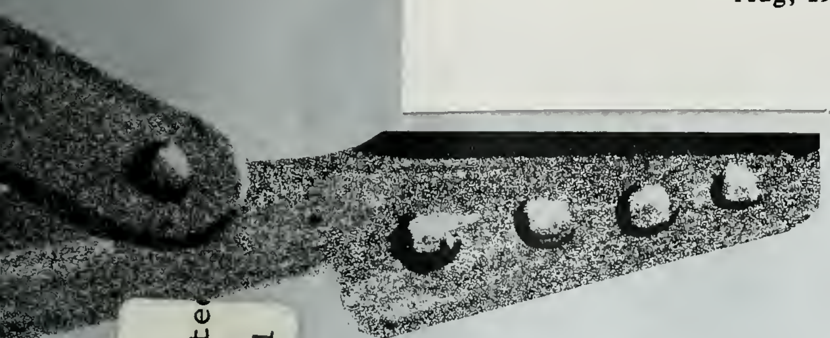


Motion Planning and Related Geometric Algorithms in Robotics

by

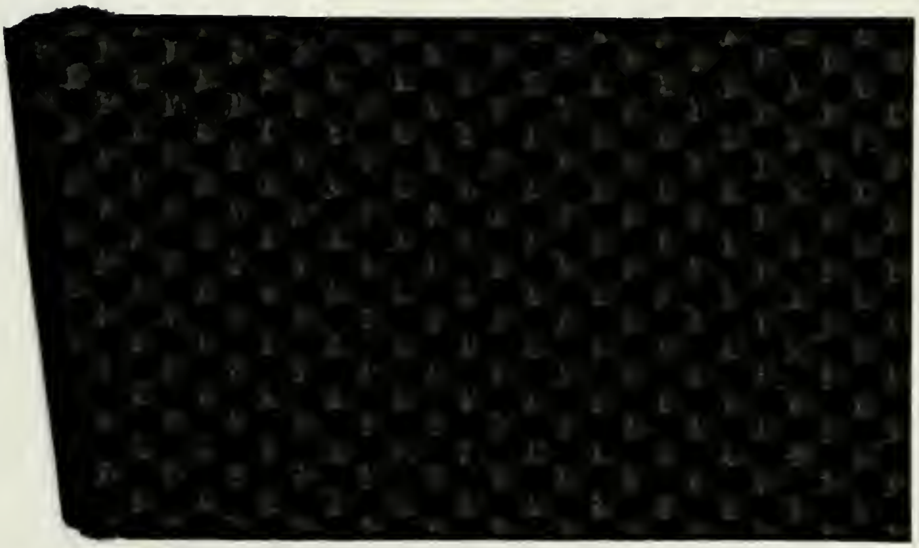
J.T. Schwartz
M. Sharir

Technical Report No. 241
Robotics Report No. 79
Aug, 1986



New York University
Courant Institute of Mathematical Sciences

Computer Science Division
251 Mercer Street New York, N.Y. 10012



Motion Planning and Related Geometric Algorithms in Robotics

by

J.T. Schwartz
M. Sharir

Technical Report No. 241
Robotics Report No. 79
Aug, 1986

New York University
Dept. of Computer Science
Courant Institute of Mathematical Sciences
251 Mercer Street
New York, New York 10012

School of Mathematical Sciences
Tel Aviv University

Work on this paper has been supported by Office of Naval Research Grant N00014-82-K-0381, National Science Foundation CER Grant DCR-83-20085, and by grants from the Digital Equipment Corporation and the IBM Corporation.

Motion Planning and Related Geometric Algorithms in Robotics

J.T. Schwartz and M. Sharir

Courant Institute of Mathematical Sciences
New York University
and
School of Mathematical Sciences
Tel Aviv University

1. Introduction

Research on theoretical problems in robotics looks ahead to a future generation of robots substantially more autonomous than present robotic systems, whose current algorithmic and software capabilities are rather primitive.

The improved capabilities which this research aims to create can be grouped into three broad categories: sensing, planning, and control.

Of these three, planning involves the use of an environment model to carry out significant parts of a robot's activities automatically. The aim is to allow the robot's user to specify a desired activity in very high level, general terms, and then have the system fill in the missing low-level details. For example, the user might specify the end product of some assembly process, and ask the system to construct a sequence of assembly substeps; or, at a less demanding level, to plan collision-free motions which pick up individual subparts of an object to be assembled, transport them to their assembly position, and insert them into their proper places.

Studies in this area have shown it to have significant mathematical content; tools drawn from classical geometry, topology, algebraic geometry, algebra, and combinatorics have all proved relevant. This work relates closely to work in computational geometry, an area which has also progressed very rapidly during the last few years.

2. Statement of the problem

In its simplest form, the motion planning problem can be defined as follows. Let B be a robot system consisting of a collection of rigid subparts (some of which may be attached to each other at certain joints while others may move independently) having a total of k degrees of freedom, and suppose that B is free to move in a two or three dimensional space V amidst a collection of obstacles whose geometry is known to the robot system. The *motion planning problem* for B is: Given an initial position Z_1 and a desired final position Z_2 of B, determine whether there exists a continuous obstacle-avoiding motion of B from Z_1 to Z_2, and if so plan such a motion. See Fig. 1 for an illustration of this problem.

This problem has been studied in many recent papers (cf. [LPW], [Mo1], [Ud], [Re], [SS1], [SS2], [SS3], [SA], [SS4], [Ya1], [OY], [OSY1], [OSY2], [LS1], [LS2], [LS3], [KS1], [KS2], [KLPS], [SiS], [HJW1], [HJW2], [HW], [HSS2], [JP], [Ya3], [Ya4]). Interesting heuristic and approximating approaches to the problem have also been developed by Lozano-Perez, Brooks, Mason, Taylor and their collaborators; see [Br1], [Br2], [BLP], [LP], [LP2]. It is equivalent to the problem of calculating the path-connected components of the (k-dimensional) space FP of all *free positions* of B (i.e. the set of positions of B in which B does not contact any obstacle), and is therefore a problem in "computational topology". In general FP is a high-dimensional space with irregular boundaries, and is thus hard to calculate efficiently.

Section 3 of this survey will discuss the problem's inherent complexity, detail several techniques developed for treating it, and review various efficient algorithms for planning the motions of certain special robot systems having only few degrees of freedom.

An interesting extension of the motion planning problem is to the case in which the environment contains objects moving in some known and predictable manner. Although this problem has been little studied, the few results obtained so far seem to indicate that it is inherently harder than the static problem.

All the problem variants mentioned so far aim to determine whether a collision-free path exists between two specified system positions, and, if so, to produce *some* such path. A further issue is to produce a path which satisfies some criterion of optimality. For example, if a mobile robot is approximated as a single moving point, one might want to find the shortest Euclidean path between the initial and final system positions. In more complex situations the notion of optimal motion is less clearly defined, and has as yet been little studied.

Studies of the motion planning problem tend to make heavy use of many algorithmic techniques in computational geometry. Various motion-planning related problems in computational geometry will be reviewed in Section 5.

3. Motion Planning in Static and Known Environments.

As above, let B be a moving robot system, k be its number of degrees of freedom, V denote the two or three dimensional space in which B is free to move, and FP denote the space of free positions of B, as defined above. The space FP is determined by the collection of algebraic inequalities which express the fact that at position Z the system B avoids collision with any of the obstacles present in its workspace. We will denote by n the number of inequalities needed to define FP, and call it the "geometric (or combinatorial) complexity" of the given instance of the motion planning problem. As noted, we make the reasonable assumption that the parameters describing the degrees of freedom of B can be chosen in such a way that each of these inequalities is algebraic. Indeed, the group of motions (involving various combinations of translations and rotations) available to a given robot can ordinarily be given algebraic representation, and the system B and its environment V can typically be modeled as objects bounded by a collection of algebraic surfaces (e.g., polyhedral, quadratic, or spline-based).

(a) The general motion planning problem.

Assuming then that FP is an algebraic or semi-algebraic set in E^k, it was shown in Schwartz and Sharir [SS2] that the motion planning problem can be solved in time polynomial in the number n of algebraic constraints defining FP and in their maximal degree, but doubly exponential in k. The general procedure described uses a decomposition technique due to Collins [Co] and originally applied to Tarski's theory of real closed fields.

Collins' key definitions and theorems are as follows:

Definition 1. For any subset X of Euclidean space, a *decomposition* of X is a finite collection K of disjoint connected subsets Y of X whose union is X. Such a decomposition is a *Tarski decomposition* if each subset Y is a Tarski set, i.e. a ('semi-algebraic') set described by finitely many polynomial equalities and inequalities.

Definition 2. A *cylindrical algebraic decomposition* of E^r is defined as follows. For $r=1$ such a decomposition is just a partitioning of E^1 into a finite set of algebraic numbers and into the finite and infinite open intervals delimited by these numbers. For $r>1$, a cylindrical algebraic decomposition

of E^r is a decomposition K obtained recursively from some cylindrical algebraic decomposition K' of E^{r-1} as follows. Regard E^r as the Cartesian product of E^{r-1} and E^1 and accordingly represent each point p of E^r as a pair $[x,y]$ with $x \in E^{r-1}$ and $y \in E^1$. Then K must be defined in terms of K' and an auxiliary polynomial $P=P(x,y)$ with rational coefficients, in the following way:

(i) For each $c \in K'$, let $c \times E^1$ designate the *cylinder over c*, i.e., the set of all $[x,y]$ such that $x \in c$.

(ii) For each $c \in K'$ there must exist an integer n, such that for each $x \in c$ there are exactly n distinct real roots $f_1(x), \ldots, f_n(x)$ of $P(x,y)$ (regarded as a polynomial in y), and these roots must vary continuously with x. We suppose in what follows that these roots have been enumerated in ascending order. Then each one of the cells of K which intersects $c \times E^1$ must have one of the following forms:

(ii.a) $\{[x,y] : x \in c, y<f_1(x)\}$ (lower semi-infinite "segment" of $c \times E^1$).

(ii.b) $\{[x, f_i(x)] : x \in c\}$ ("section" of $c \times E^1$).

(ii.c) $\{[x,y] : x \in c, f_i(x)<y<f_{i+1}(x)\}$ ("segment" of $c \times E^1$).

(ii.d) $\{[x,y] : x \in c, f_n(x)<y\}$ (upper semi-infinite "segment" of $c \times E^1$).

All these cells are said to have c as their *base cell* in K'; K' is said to be the *base decomposition*, and P the *base polynomial*, of K.

It follows easily by induction that each of the sets constituting a cylindrical algebraic decomposition K of E^r is topologically equivalent to an open cell of some dimension $k \le r$. We can therefore refer to the elements $c \in K$ as the (open) *Collins cells of the decomposition K*.

Definition 3. Let S be a set of functions of r variables, and K a cylindrical algebraic decomposition of E^r. Then K is said to be $S-$*invariant* if, for each c in K and each f in S, one of the following conditions holds uniformly for $x \in c$: either

(a) $f(x)=0$ for all $x \in c$; or

(b) $f(x)<0$ for all $x \in c$; or

(c) $f(x)>0$ for all $x \in c$.

Definition 4. A point $p \in E^r$ is *algebraic* if each of its coordinates is a real algebraic number. A *defining polynomial* for p is a polynomial with rational coefficients whose set of roots includes all the coordinates of p.

Theorem 1 (Collins). Given any finite set S of polynomials with rational coefficients in r variables, one can effectively construct an S-invariant cylindrical algebraic decomposition K of E^r into Tarski sets such that each

$c \in K$ contains an algebraic point. Moreover, defining polynomials for all these algebraic points, and quantifier-free defining formulae for each of the sets $c \in K$, can also be constructed effectively.

The theorem is proved by inductive consideration of polynomial resultants and is hence constructive (though by no means suggestive of an efficient algorithm). [SS2] supplements this result with the following technical definition and addendum:

Definition 5. A Collins decomposition K is said to be *well−based* if the following condition holds. Let K' be the base decomposition and $P(b,x)$ the base polynomial of K. Then we require that $P(b,x)$ should not be identically zero for any $b \in E^{r-1}$. Moreover, we require that this same condition apply recursively to the base decomposition K'.

Theorem 2. The collection of compact cells of a (well-based) Collins decomposition of E^r forms a regular cell complex.

Corollary 3. For each j, the (singular) homology group $H_j(V)$ of the real algebraic variety V defined by any set Π of polynomial equations $P(x_1, \ldots, x_n)=0$ with rational coefficients can be computed in a purely rational manner from the coefficients of the polynomials P.

The techniques used to prove this theorem relate closely to the method used by Hironaka [Hi] to prove triangulability of real algebraic variables. Constructive determination of the cell incidence relationships needed to prove Corollary 3 is achieved by considering the Laurent series for solutions of polynomial equations derived from the cell decomposition.

Corollary 3 yields an effective procedure for calculating the 0-th homology group, i.e. the connected components, of any semi-algebraic configuration space FP. The complexity of this procedure is shown in [SS2] to be of the same order of magnitude of Collins' original procedure, and, as stated above, is polynomial in the number of constraints defining FP and in their maximal degree, but is doubly exponential in k. Though hopelessly inefficient in practical terms, this result nevertheless serves to calibrate the computational complexity of the motion-planning problem.

(b) Lower bounds

The result just cited suggests that motion planning becomes harder rapidly as the number k of degrees of freedom increases; this conjecture has in fact been proved for various model 'robot' systems. Specifically, Reif [Re] proved that motion planning is PSPACE-hard for a certain 3-D system involving arbitrarily many links and moving through a complex system of narrow tunnels. Since then PSPACE-hardness has been established for

simpler moving systems, including 2-D systems of mechanical linkages (Hopcroft, Joseph, and Whitesides [HJW2]), a system of 2-D independent rectangular blocks sliding inside a rectangular box (Hopcroft, Schwartz and Sharir [HSS2]), and a single 2-D arm with many links moving through a 2-D polygonal space (Joseph and Plantinga [JP]). Several weaker results establishing NP-hardness for still simpler systems have also been obtained.

The Hopcroft-Joseph-Whitesides result is established by showing that, given an arbitrary Turing machine T with a fixed bounded tape memory, one can construct a planar linkage L whose motions simulate the actions of T, so that L can only move from a specified initial to a specified final configuration if the Turing machine T eventually halts. The size of the linkage L constructed is polynomially bounded by the size of T's state table and memory tape. One proceeds by noting that the actions of an arbitrary T can easily be characterized by a set of polynomial constraints, and then by using the (classical; cf. [Ke]) result which shows how to construct a mechanical linkage capable of representing any specified multivariate polynomial $P(x_1, \ldots, x_n)$.

In more detail, a *planar linkage* is a mechanism consisting of finitely many rigid rods, of prespecified lengths, joined together at some of their endpoints by hinge-pins about which they are free to rotate. Any number of rod-ends are allowed to share a common hinge-pin; and particular pins can be held at specified points by being 'fastened to the plane.' Aside from this, the hinge-pins and rods are free to move in the plane, and it is assumed that the motion of one rod does not impede the motion of any other (i.e. the rods are allowed to 'pass over' each other). Such a linkage is said to represent the multivariate polynomial $P(x_1, \ldots, x_n)$ if there exist n hinge pins $p_1, \ldots, p_n$ which the linkage constrains to move along the real axis, and an $n+1$-st hinge pin p_0 which the linkage constrains to lie at the real point $P(x_1, \ldots, x_n)$ whenever $p_1, \ldots, p_n$ are placed at the real points $x_1, \ldots, x_n$. (It is assumed that the linkage leaves $p_1, \ldots, p_n$ free to move independently over some large interval of the real axis.)

The existence of a linkage representing an arbitrary polynomial P in the case just explained is established by exhibiting linkages which realize the basic operations of addition, multiplication, etc. and then showing how to represent arbitrary combinations of these operations by fastening sublinkages together appropriately.

The Hopcroft-Schwartz-Sharir result on *PSPACE*-hardness of the coordinated motion planning problem for an arbitrary set of rectangular blocks moving inside a rectangular frame is proved similarly. It is relatively easy to show that the actions of an arbitrary tape-bounded Turing machine

can be imitated by the motions of a collection of similarly-sized nearly rectangular 'keys' whose edges bear protrusions and indentations which constrain the manner in which these 'keys' can be juxtaposed, and hence the manner in which they can move within a confined space. A somewhat more technical discussion then shows that these keys can be cut appropriately into rectangles without introducing any significant possibilities for motion of their independent parts that do not correspond to motions of entire keys.

(c) The "projection method".

In spite of these negative worst-case results, algorithms of varying levels of efficiency for planning the motions of various simple robot systems have been developed. These involve several general approaches to the design of motion planning algorithms. The first such approach, known as the *projection method,* uses ideas similar to those appearing in the Collins decomposition procedure described above. One fixes some of the problem's degrees of freedom (for the sake of exposition, suppose just one parameter y is fixed, and let $\bar{x}$ be the remaining parameters); then one solves the resulting restricted $k-1$-dimensional motion planning problem. This subproblem solution must be such as to yield a discrete combinatorial representation of the restricted free configuration space (essentially, a cross-section of the entire space FP) that changes only at a finite collection of 'critical' values of the final parameter y. These critical values of y are then calculated; they partition the entire space FP into connected cells, and by calculating relationships of adjacency between these cells one can describe the connectivity of FP by a discrete *connectivity graph* CG. This graph has the aforesaid cells as vertices, and has edges which represent relationships of cell adjacency in FP. The connected components of FP correspond in a one-to-one manner to the connected components of CG, reducing the problem to a discrete path searching problem in CG.

This relatively straightforward technique was applied in a series of papers by Schwartz and Sharir on the "piano movers" problem, to yield polynomial time motion planning algorithms for various specific systems, including a rigid polygonal object moving in 2-D polygonal space [SS1], two or three independent discs moving in coordinated fashion in 2-D polygonal space [SS3], certain types of multi-arm linkages moving in 2-D polygonal space [SA], and a rod moving in 3-D polyhedral space [SS4]. These initial solutions were coarse and not very efficient; subsequent refinements have improved their efficiency substantially.

A typical example that has been studied extensively is the case of a line segment B (a "rod") moving in two-dimensional polygonal space whose boundary consists of n segments ("walls"). Here the configuration space FP

is three-dimensional, and it can be decomposed into cells efficiently using a modified projection technique developed by Leven and Sharir [LS1].

In this approach one starts by restricting the motion of B to a single degree of freedom of translation along its length. For this trivial subproblem the restricted *FP* simply consists of an interval which can be represented by a discrete label $[w_1, w_2]$ consisting of the two walls against which B stops when moving backwards or forwards from its given placement.

Next one admits a second degree of freedom by allowing arbitrary translational motion of B. Assuming that B points in the direction of the positive y-axis, the second degree of freedom is parameterized by the x-coordinate of B. As x varies, the label $[w_1, w_2]$ of the placements of B remains unchanged until one reaches either an endpoint of w_1 or of w_2, or a wall corner lying between w_1 and w_2, or a placement in which B simultaneously touches both w_1 and w_2. Hence, given an initial placement Z of B with label $[w_1, w_2]$, we can define a 2-D "non-critical" region R consisting of all placements of B which are reachable from Z by a translational motion during which the label does not change. R itself can be uniquely represented by another discrete label of the form $\lambda(R)=[w_1, w_2, LEFT, RIGHT]$, where *LEFT*, *RIGHT* describe the type of criticality (an endpoint of w_1 or of w_2, a new corner between w_1 and w_2, or a "dead-end" at which B gets stuck) defining respectively the left and right boundaries of R.

Finally one introduces the final rotational degree of freedom θ. Again one can show that the label $\lambda(R)$ of the non-critical region R containing the initial placement Z of B remains constant as θ varies, unless θ crosses a *critical orientation* at which the left or right boundary of R either (i) comes to contain two wall corners; or (ii) contains an endpoint of w_1 or of w_2 and the distance from that endpoint to the other wall (in the direction of θ or $\theta+\pi$) is equal to the length of B; or (iii) contains another corner and the distance between w_1 and w_2 along this boundary is equal to the length of B. See Fig. 2 for an illustration of these critical conditions.

One can therefore define a 3-D non-critical cell C of *FP* containing Z to consist of all placements of B reachable from Z by a motion during which the label of the 2-D region enclosing the placement of B remains constant, and represent each such cell by a triple $[\lambda, \theta_1, \theta_2]$, where λ is the label of the 2-D region enclosing Z and where θ_1, θ_2 are two critical orientations at which the label λ changes discontinuously. The collection of these cells yields the desired partitioning of *FP*.

Leven and Sharir show that the number of critical orientations is at most $O(n^2)$, and that, assuming B and the walls are in "general position", each

critical orientation delimits only a small constant number of cells. Thus the total number of cells in FP is also $O(n^2)$. [LS1] presents a fairly straightforward algorithm, which runs in time $O(n^2 \log n)$, for constructing these cells and for establishing their adjacency in FP, a very substantial improvement of the $O(n^5)$ algorithm originally presented in [SS1].

O'Rourke [OR] has recently shown that for certain configurations there exist two placements of B reachable from one another, but are such that any motion between them must consist of $\Omega(n^2)$ different simple submotions, proving that the Leven-Sharir algorithm is nearly optimal in the worst case.

(d) The "retraction method" and other approaches to the motion-planning problem.

Several other important algorithmic motion planning techniques were developed subsequent to the simple projection technique originally considered. The so-called *retraction method* proceeds by retracting the configuration space FP onto a lower dimensional (usually a 1-dimensional) subspace N, so that two system positions in FP lie in the same connected component of FP if and only if their retractions to N lie in the same connected component of N. This reduces the dimension of the problem, and if N is 1-dimensional the problem becomes one of searching a graph.

O'Dunlaing and Yap [OY] introduced this retraction technique in the simple case of a disc moving in 2-D polygonal space. Here the subspace N can be taken to be the Voronoi diagram associated with the set of given polygonal obstacles. Their technique yields an $O(n \log n)$ motion-planning algorithm. After this first paper O'Dunlaing, Sharir and Yap [OSY1], [OSY2] generalized the retraction approach to the case of a rod moving in 2-D polygonal space by defining a variant Voronoi diagram in the 3-D configuration space FP of the rod, and by retracting onto this diagram. This diagram consists of all placements of the rod at which it is simultaneously nearest to at least two obstacles. The Voronoi diagram defined by a set of obstacles in general position can readily be divided into 2-D Voronoi sheets (placements in which the rod is simultaneously nearest to two obstacles), which are bounded by 1-D Voronoi edges (placements in which the rod is nearest to three obstacles), which in turn are delimited by Voronoi vertices (placements in which the rod is nearest to four obstacles; cf. Fig. 3). The algorithm described in [OSY1], [OSY2] actually constructs a 1-D subcomplex within the Voronoi diagram; this complex consists of the Voronoi edges and vertices plus some additional connecting arcs. It is shown in [OSY1] that this Voronoi "skeleton" characterizes the connectivity of FP, in the sense that each connected component of FP contains exactly one connected component of the skeleton. A fairly involved geometric analysis given in [OSY2] shows

that the total number of Voronoi vertices is $O(n^2\log^* n)$, and that the entire "skeleton" can be calculated in time $O(n^2\log n \log^* n)$ (a substantial improvement of the original projection technique, but nevertheless a result shortly afterward superceded by [LS1]).

A similar retraction approach was used by Leven and Sharir [LS2] to obtain an $O(n \log n)$ algorithm for planning the purely translational motion of a simple convex object B amidst polygonal barriers. This last result uses another generalization of Voronoi diagram, known as the B-Voronoi diagram, which is defined as follows. Let O be a reference point within B, which we assume to lie initially at the origin, and define a generalized distance function d_B by

$$d_B(p,q) = \min \{\lambda : q \in p+\lambda B\} .$$

(The generalized distance function d_B satisfies the triangle inequality but is not symmetric in general.) Define the B-Voronoi diagram of the given set S of obstacles to consist of all points p for which there exist at least two obstacles s_1, s_2 such that

$$d_B(p,s_1) = d_B(p,s_2) \leq d_B(p,s)$$

for all $s \in S$; see Fig. 4 for an illustration of such a diagram.

Though of a more complex structure than standard Voronoi diagrams, the B-diagram retains most of the useful properties of standard Voronoi diagrams. In particular, its size is linear in the number of obstacles in S, and, if B has sufficiently simple shape, can be calculated in time $O(n \log n)$, using a variant of the technique described in [Ya2].

Next, let N be the portion of the B-diagram consisting of points whose B-distance from the nearest obstacle is greater than 1. Then any translate of B in which the reference point O on B is placed at a point in N is a free placement of B. It is proved in [LS2] that N characterizes the connectivity of the free configuration space of B, in the sense defined above, so that, for purpose of planning, motion of B can be restricted to have the reference point O move only along N. This yields an $O(n \log n)$ motion planning algorithm for this case. (A somewhat simpler $O(n \log^2 n)$ algorithm, based on a general technique introduced by Lozano-Perez and Wesley [LPW], was described somewhat earlier by Kedem and Sharir [KS1] (cf. also [KLPS]); this last result makes use of an interesting topological property of intersecting planar Jordan curves.)

Recently Sifrony and Sharir [SiS] have devised another retraction-based algorithm for the motion of a rod in 2-D polygonal space. Their approach is to construct the 1-D network of arcs within FP consisting of all the 1-D edges on the boundary of FP (each such edge consists of semi-free placements in

which the rod simultaneously makes two specific contacts with the obstacles), plus several additional arcs which connect different components of the boundary of *FP* that bound the same connected component of *FP*. Again, this network characterizes the connectivity of *FP*, so that a motion planning algorithm need only consider motions proceeding within this network. The Sifrony-Sharir approach generates motions in which the rod is in contact with the obstacles, and is thus conceptually somewhat inferior to the Voronoi-diagram based techniques, which aim to keep the moving system between obstacles, not letting it get too close to any single obstacle. However the network in [SiS] is much simpler to analyze and to construct. Specifically, it is shown in [SiS] that this network has $O(n^2)$ vertices and edges and can be constructed in time $O(n^2 \log n)$. (Actually, the network size is bounded by the number K of pairs of obstacles lying at distance less than or equal to the length of the moving rod, and the complexity of the algorithm is bounded by $O(K \log n)$. Thus if the obstacles are not too badly cluttered together, the Sifrony-Sharir algorithm will run quite efficiently; this makes the approach in [SiS] more attractive than the previous solutions in [OSY2], [LS1].)

Hybrid techniques are applicable for certain cases of motion planning. For example, in an analysis of the motion planning problem for a convex polygonal object moving in 2-D polygonal space, Kedem and Sharir [KS2] obtained an $O(n^2 \beta(n) \log n)$ motion planning algorithm (where $\beta(n)$ is a very slowly growing function of n) using a hybrid approach which involves projection of *FP* onto a 2-D space in which the orientation θ of the object is fixed, followed by retraction of this 2-D space roughly onto its boundary. This result makes use of a combinatorial result of Leven and Sharir [LS3], reviewed later in this paper.

4. Variants of the Motion Planning Problem

(a) Optimal motion planning

The only optimal motion planning problem which has been studied extensively thus far is that in which the moving system is represented as a single point, in which case one aims to calculate the shortest Euclidean path connecting initial and final system positions, given that specified obstacles must be avoided. Most existing work on this problem assumes that the obstacles are either polygonal (in 2-space) or polyhedral (in 3-space).

The 2-D case is considerably simpler than the 3-D case; see Fig. 5. When the free space V in 2-D is bounded by n straight edges, it is easy to calculate the desired shortest path in time $O(n^2 \log n)$. This is done by constructing a *visibility graph VG* whose edges connect all pairs of boundary corners of V

which are visible from each other through V, and then by searching for a shortest path through VG (see [ShS] for a sketch of this idea). This procedure was improved to $O(n^2)$ by Asano et al. [AAGHI], by Welzl [We], and by Reif and Storer [RS], using a cleverer method for constructing VG. Their quadratic time bound has been improved in certain special cases. However, it is not known whether shortest paths for a general polygonal space V can be calculated in sub-quadratic time. Among the special cases allowing more efficient treatment the most important is that of calculating shortest paths inside a simple polygon P. Lee and Preparata [LeP] gave a linear time algorithm for this case, assuming that a triangulation of P is given in advance. The Preparata-Lee result was recently extended by Guibas et al. [GHLST], who gave a linear time algorithm which calculates all shortest paths from a fixed source point to all vertices of (a tringulated polygon) P.

Other results on 2-D shortest paths include an $O(n \log n)$ algorithm for finding *rectilinear* shortest paths which avoid n rectilinear disjoint rectangles [dRLW]; an $O(n^2 \log n)$ algorithm for finding Euclidean shortest motion of a circular disc in a 2-D polygonal region [Ch]; algorithms for cases in which the obstacles consist of a small number of disjoint convex regions [RS]; algorithms for the 'weighted region' case (in which the plane is partitioned into polygonal regions in each of which the path has a different multiplicative cost factor) [MP]; and various other special cases.

The 3-D polyhedral shortest path problem is substantially more difficult. To calculate shortest paths in 3-space amidst polyhedral obstacles bounded by n edges, we can begin by noting that any such path must be piecewise linear, with corners lying on the obstacle edges, and that it must subtend equal incoming and outgoing angles at each such corner; see Fig. 6. These remarks allow shortest path calculation can to be split into two subproblems: (i) Find the sequence of edges through which the desired path must pass; (ii) Find the contact points of the path with these edges. However, even when the sequence of edges crossed by the path is known, calculation of the contact points is still non-trivial, because it involves solution of a large system of quartic equations, expressing the conditions of equal incoming and outgoing angles at each crossed edge, whose precise solution may require use of the previously described Collins procedure, thus requiring time doubly exponential in n. (Note however that Reif and Storer [RS] have reduced this time to singly exponential by a clever re-use of variables in these equations, reducing the number of variables to $O(\log n)$ only.) Subproblem (ii) can also be solved using numerical iterative procedures, but even if this approximating approach is taken there still remains subproblem (i), whose only known solution to date is exhaustive enumeration of all possible edge sequences. A recent result of Canny and Reif [CR] indicates that the 3-D problem is indeed

NP-hard.

One of the reasons the problem is difficult in the general polyhedral case is that consecutive edges crossed by the shortest path can be skewed to one another. There are, however, some special cases in which this difficulty does not arise, and they admit efficient solutions. One such case is that in which we aim to calculate shortest paths lying along the surface of a single convex polyhedron having n edges. In this case subproblem (ii) can easily be solved by 'unfolding' the polyhedron surface at the edges crossed by the path, thereby transforming the path to a straight segment connecting the unfolded source and destination points (cf. [ShS]). Extending this observation, Mount [Mo2] has devised an $O(n^2 \log n)$ algorithm, which proceeds by sophisticating an algorithmic technique originally introduced by Dijkstra to find shortest path in graphs, specifically by maintaining and updating a combinatorial structure characterizing shortest paths from a fixed initial point to each of the edges of the polyhedron (cf. also [ShS] for an initial version of this approach).

This result has recently been extended in several ways. A similar $O(n^2 \log n)$ algorithm for shortest paths along a (not necessarily convex) polyhedral surface is given in [MMP]. [BS] consider the problem of finding the shortest path connecting two points lying on two disjoint convex polyhedral obstacles, and report a nearly cubic algorithm, which makes use of the Davenport-Schinzel sequences described below. The case of shortest paths which avoid a fixed number k of disjoint convex polyhedral obstacles is analyzed in [Sh3], which describes an algorithm that is polynomial in the total number of obstacle edges, but is exponential in k. Finally, an approximating pseudo-polynomial scheme for the general polyhedral case is reported in [Pa]; this involves splitting each obstacle edge by adding sufficiently many new vertices and by searching for the shortest piecewise linear path bending only at those vertices.

(b) Adaptive and exploratory motion planning

If the environment of obstacles through which motion must be planned is not known to a robot system a priori, but the system is equipped with sensory devices, motion planning assumes a more "exploratory" character. If only tactile (or proximity) sensing is available, then a plausible strategy might be to move along a straight line (in physical or configuration space) directly to the target position, and when an obstacle is reached, to follow its boundary until the original straight line of motion is reached again [LuS]. If vision is also available, then other possibilities need to be considered, e.g. the system can obtain partial information about its environment by viewing it from the present position, and then "explore" it to gain progressively more

information until the desired motion can be fully planned. However, problems of this sort have hardly begun to be investigated.

Even when the environment is fully known to the system, other interesting issues arise if the environment is changing. For example, when some of the objects in the robot's environment are picked up by the robot and moved to a different position, one wants fast techniques for incremental updating of the environment model and the data structures used for motion planning. Moreover, whenever the robot grasps an object to move it, robot plus grasped object become a new moving system and may require a different motion planning algorithm, but one whose relationship to motions of the robot alone needs to be investigated. Adaptive motion planning problems of this kind have hardly been studied as yet.

(c) Motion planning in the presence of moving obstacles

Interesting generalizations of the motion planning problem arise when some of the obstacles in the robot's environment are assumed to be moving along known trajectories. In this case the robot's goal will be to "dodge" the moving obstacles while moving to its target position. In this "dynamic" motion planning problem, it is reasonable to assume some limit on the robot's velocity and/or acceleration. Two initial studies of this problem by Reif and Sharir [RSh] and by Sutner and Maass [SM] indicate that the problem of avoiding moving obstacles is substantially harder than the corresponding static problem. By using time-related configuration changes to encode Turing machine states, they show that the problem is PSPACE-hard even for systems with a small and fixed number of degrees of freedom. However, polynomial-time algorithms are available in a few particularly simple special cases.

5. Results in Computational Geometry Relevant to Motion Planning

The various studies of motion planning described above make extensive use of efficient algorithms for the geometric subproblems which they involve, for which reason motion planning has encouraged research in computational geometry. Problems in computational geometry whose solutions apply to robotic motion planning include the following:

(a) Intersection detection

The problem here is to detect intersections and to compute shortest distances, e.g. between moving subparts of a robot system and stationary or moving obstacles. Simplifications which have been studied include that in which all objects involved are circular discs (in the 2-D case) or spheres (in the 3-D

case). In a study of the 2-D case of this problem, Sharir [Sh1] developed a generalization of Voronoi diagrams for a set of (possibly intersecting) circles, and used this diagram to detect intersections and computing shortest distances between discs in time $O(n \log^2 n)$ (an alternative approach to this appears in [IIM]). Hopcroft, Schwartz and Sharir [HSS1] present an algorithm for detecting intersections among n 3-D spheres which also runs in time $O(n \log^2 n)$. However, this algorithm does not adapt in any obvious way to allow proximity calculation or other significant problem variants.

Other intersection detection algorithms appearing in the computational geometry literature involve rectilinear objects and use multi-dimensional searching techniques for achieving high efficiency (see [Me] for a survey of these techniques).

(b) Generalized Voronoi diagrams

The notion of Voronoi diagram has proven to be a useful tool in the solution of many motion planning problems. The discussion given previously has mentioned the use of various variants of Voronoi diagram in the retraction-based algorithms for planning the motion of a disc [OY], or of a rod [OSY1], [OSY2], or the translational motion of a convex object [LS2], and in the intersection detection algorithm for discs mentioned above [Sh1]. The papers just cited, and some related works ([Ya2], [LS4]) describe the analysis of these diagrams and the design of efficient algorithms for their calculation.

(c) Davenport-Schinzel sequences

Davenport-Schinzel sequences are combinatorial sequences of n symbols which do not contain certain forbidden subsequences of alternating symbols. Sequences of this sort appear in studies of efficient techniques for calculating the lower envelope of a set of n continuous functions, if it is assumed that the graphs of any two functions in the set can intersect in some fixed number of points at most. These sequences, whose study was initiated in [DS], [Da], have proved to be powerful tools for analysis (and design) of a variety of geometric algorithms, many of which are useful for motion planning.

More specifically, an (n,s) Davenport Schinzel sequence is defined to be a sequence U composed of n symbols, such that (i) no two adjacent elements of U are equal, and (ii) there do not exist $s+2$ indices $i_1 < i_2 < \cdots < i_{s+2}$ such that $u_{i_1} = u_{i_3} = u_{i_5} = \cdots = a$, $u_{i_2} = u_{i_4} = u_{i_6} = \cdots = b$, with $a \neq b$. Let $\lambda_s(n)$ denote the maximal length of an (n,s) Davenport-Schinzel sequence. An early study by Szemeredi [Sz] of the maximum possible length of such sequences shows that $\lambda_s(n) \leq C_s n \log^* n$, where C_s is a constant depending on s. Improving on this result, Hart and Sharir [HS] proved that

$\lambda_3(n) = \Theta(n\,\alpha(n))$ where $\alpha(n)$ is the very slowly growing inverse of the Ackermann function. In [Sh2], [Sh5] Sharir established the bounds

$$\lambda_s(n) = O(n\alpha(n)^{O(\alpha(n)^{s-3})})$$

and

$$\lambda_s(n) = \Omega(n\,\alpha^{\lfloor (s-1)/2 \rfloor}(n))$$

for $s>3$. These results show that, in practical terms, $\lambda_s(n)$ is an almost linear function of n (for any fixed s).

Recently, numerous applications of these sequences to motion planning have been found. These include the following.

(i) Let B be a convex k-gon translating and rotating in a closed 2-D polygonal space V bounded by n edges. The *polygon containment* problem calls for determining whether there exists any free placement of B, i.e. a placement in which B lies completely within V. Some variants of this problem have been studied by Chazelle [Cha], who showed that if such a free placement of B exists, then there also exists a *stable* free placement of B, namely a placement in which B lies completely within V and makes three simultaneous contacts with the boundary of V (see Fig. 7). Using Davenport-Schinzel sequences, Leven and Sharir have shown in [LS3] that the number of such free stable placements is at most $O(kn\lambda_6(kn))$, and that they can all be calculated in time $O(kn\lambda_6(kn) \log kn)$. Thus, within the same time bound, one can determine whether P can be placed inside Q.

The analysis of [LS3] proceeds as follows. There are $O(kn)$ possible edge-vertex or vertex-edge contacts of B against the boundary Q of V. Fix such a contact C, e.g. of a vertex S of B against an edge W of Q, and consider the space $F = F_C$ of all (not necessarily free) positions of B in which S touches W. F is a two-dimensional space parameterized by the displacement x of S along W and by the orientation θ of B. For each additional contact C_i we consider the (1-D) locus γ_{C,C_i} of placements of B in F_C in which the contact C_i is also made. Assume that B and Q are in general position (so that no degenerate multiple contacts of P against Q can occur; see [LS3] for more detail). Then we have

(a) For each pair of contacts C_i,C_j, the curves γ_{C,C_i} and γ_{C,C_j} intersect in at most 4 points. This property actually states that three fixed contacts of B against Q can occur simultaneously in at most four placements of B.

(b) Each curve $\gamma_{C,C'}$ consists of at most five connected smooth portions.

(c) For each fixed orientation θ_0 in which B can make two contacts C,C' simultaneously, either all placements (x,θ_0) of P in F_C lying on one specific side of $\gamma_{C,C'}$ are non-free, or all placements (x,θ_0) of P in $F_{C'}$ lying on one

specific side of $\gamma_{C',C}$ are non-free.

For each contact C we let $\psi_C^-(\theta)$ (resp. $\psi_C^+(\theta)$) denote the lower (resp. upper) envelope (in the $\theta-x$ space) of all the curves $\gamma_{C,C'}$ having the property that all points (θ,x) lying above (resp. below) $\gamma_{C,C'}$ represent non-free placements of B. Property (c) is then seen to imply that each free stable placement of B (in which it makes three simultaneous contacts C_1, C_2, C_3) must be represented in one of the following ways:

(i) By a *breakpoint* on one of the envelopes $\psi_{C_i}^-(\theta)$, $\psi_{C_i}^+(\theta)$, i.e. a point in which two of the curves γ_{C_i,C_j}, γ_{C_i,C_k} intersect along the corresponding envelope.

(ii) By an intersection of the upper and the lower envelopes within one of the spaces F_{C_i}.

(iii) By three points, lying respectively on either the upper or the lower envelope of F_{C_i}, for $i=1,2,3$.

Stable placements of type (i) are easy to find by the standard divide and conquer technique [At] for calculation the lower envelope of a given collection of functions. The theory of Davenport-Schinzel sequences imply, using properties (a), (b), that the total number of breakpoints along a single envelope is at most $\lambda_6(kn)$ and that they can all be calculated in time $O(\lambda_6(kn)\log kn)$. Thus the total number of type (i) placements is $O(kn\lambda_6(kn))$. Similar bounds are also easy to derive for type (ii) placements. Type (iii) placements, which are the most complex to analyze, are calculated by a priority-queue based technique which obtains these placements incrementally by stepping through the combined list of all critical orientations at which type (i) placements can arise. More detail can be found in [LS3].

Based on this result, Kedem and Sharir [KS2] have produced an $O(kn\lambda_6(kn)\log kn)$ algorithm for planning the motion of a convex k-gon B in a 2-D polygonal space bounded by n edges. Their technique proceeds as follows.

Let Q be the boundary of the polygonal region V in which B is free to move (translate and rotate) and which is bounded by n edges. Let F be the 3-D free configuration space of B (consisting of all free placements of B within V). Each placement in F is parametrized by (x,y,θ), where (x,y) is the position of some reference point on B, and θ is the orientation of B.

We first hold θ fixed, and consider the 2-D cross-section F_θ of F consisting of all free placements of B within V at orientation θ. The results in [KS], [LS2], [KLPS] imply that F_θ is a polygonal region bounded by at most $O(kn)$ edges, and that it can be calculated in time $O(kn\log kn)$ using generalized Voronoi diagrams (or by a simpler $O(kn\log^2 kn)$ algorithm).

The next and main step is to obtain a combinatorial representation of the boundary of F in a way that enables one to group together all boundary features (corners, curvilinear edges and faces) which bound the same connected component of F. This is done by constructing an *edge graph* EG whose nodes are the 1-D edges on the boundary of F (each such edge is a maximal connected portion of the locus of free placements of B in which it makes two specific obstacle contacts simultaneously), and whose edges represent adjacency of these F-edges along the boundary of F. More precisely, two edges e_1 and e_2 are said to be adjacent along the boundary of F if there exists some θ such that both e_1 and e_2 intersect the cross section F_θ of F at two corners that are adjacent corners of some connected component of F_θ.

It is proved that (assuming that B and Q are in "general position"; cf. [KS2]) two edges e_1, e_2 of F lie on the boundary of the same connected component of F if and only if their corresponding nodes in EG lie in the same connected component of that graph. Hence, given any two free placements Z_1, Z_2 of B, these placements can first be "retracted" towards the boundary of F until two corresponding placements lying on two edges e_1, e_2 of F are reached, following which one can check whether these two edges lie in the same connected component of EG. If so, it is easy to devise a straightforward procedure for transforming the path in EG connecting these two edges into a continuous motion in F between the two retracted placements of B.

The complexity of this procedure depends on the complexity of EG and on the time needed to construct that graph. It can be shown that EG has $O(kn\, \lambda_6(kn))$ vertices and edges, and that they can be all calculated in time $O(kn\, \lambda_6(kn) \log kn)$. This is done by calculating *critical orientations* θ in which the combinatorial representation of the cross section F_θ of F changes. These are shown to be orientations θ at which either a stable placement of B occurs, or at which a convex corner of F_θ comes to lie directly below a convex or non-convex corner. This additional type of critical orientations can be analyzed by a technique similar to that used in the polygon containment algorithm described above; [Ks2] gives bounds on the number of such critical orientations and on the time needed to calculate them which are similar to those obtained above for stable placements.

Having obtained all these critical orientations, one next constructs the graph EG by "sweeping" a cross section F_θ across F, updating it incrementally each time we cross through a critical orientation, and constructing on-the-fly nodes and edges of EG during that sweep. More details are given in [KS2].

Two additional recent applications of Davenport-Schinzel sequences are:

(ii) an $O(mn\,\alpha(mn) \log m \log n)$ algorithm for separating two interlocking simple polygons by a sequence of translations [PSS], where it is assumed that the polygons have m and n sides respectively.

(iii) an $O(n^2 \lambda_{10}(n) \log n)$ algorithm for finding the shortest Euclidean path between two points in 3-space avoiding the interior of two disjoint convex polyhedra having n faces altogether [BS].

Other applications are found in [At], [HS], [CS], [SL], [OSY2].

(d) Topological results related to motion planning.

Hopcroft and Wilfong [HW] have derived interesting qualitative results concerning the motion planning problem using ideas drawn from homology theory. Their basic idea is roughly as follows. Consider a connected rigid planar body B moving in the complement of a connected obstacle O. Both the body and the obstacle are assumed to be closed bounded regions. The space S of all positions of the body is defined by two translational and one rotational parameter, and can be viewed as a topological equivalent of 3-dimensional Euclidean space provided that we distinguish between otherwise identical body positions differing from each other by 360 ° rotations (i.e. pass to the simply connected covering space of the ordinary space of positions of B.) Let X be the (closed) set of all positions in which B contacts O (possibly overlapping O, i.e. with contact between the interior of B and the interior of O), and let Y be the (also closed) set of positions in which B does not overlap O (but contact between the boundaries of B and of O is allowed.) Then plainly $S = X \cup Y$, while $X \cap Y$ is the space of all positions in which B contacts, but does not penetrate, O.

The set X is easily seen to be connected. Indeed, take some point $x_0 \in B$, and consider a body position $p_1 \in X$ for which some other point $x_1 \in B$ also belongs to O. Then since both B and O are connected, a translational motion of B connects the position p_1 to a body position p_0 in which x_0 lies at some standard point within O, and then by rotating B about x_0 one can bring B to a canonical position.

The relative homology groups of $H_i(S,X)$ and $H_i(Y,X \cap Y)$ are isomorphic; and since X is connected, the natural injection of $H_0(X)$ into $H_0(S)$ is an isomorphism. Hence $H_1(S,X)$ is zero, from which it follows that $H_1(Y,X \cap Y)$ is zero, as is $H_0(Y,X \cap Y) = H_0(S,X)$. Thus the natural injection $H_0(X \cap Y) \to H_0(Y)$ is an isomorphism also, i.e. two body positions p_1, p_2 in which B contacts O (without overlap) can be connected by a path along which B never overlaps O if and only if there exists such a path along which the boundaries of B and O remain continuously in contact.

Much the same argument applies in 3 dimensions, since the covering space of the space of all positions of a 3-dimensional body is topologically the product of Euclidean 3 space by a 3-sphere, whose only nontrivial homology is in dimension 3. Hopcroft and Wilfong show that these very general topological arguments extend also to robots capable of deforming as they move, explore the additional difficulties which arise if the obstacle O is not simply connected, etc.

References

[AAGHI] T. Asano, T. Asano, L. Guibas, J. Hershberger and H. Imai, Visibility polygon search and Euclidean shortest paths, *Proc. 26th IEEE Symp. on Foundations of Computer Science,* 1985, pp. 155-164.

[At] M. Atallah, Dynamic Computational Geometry, *Proc. 24th Symp. on Foundations of Computer Science,* 1983, pp. 92-99.

[BS] A. Baltsan and M. Sharir, On shortest paths between two convex polyhedra, Tech. Rept. 180, Comp. Science Dept., Courant Institute, New York University, Sept. 1985.

[Br1] R.A. Brooks, Solving the find-path problem by good representation of free space, *IEEE Trans. Sys. Man & Cyber.* SMC-13(1983) (3), pp. 190-197.

[Br2] R.A. Brooks, Planning collision-free motions for pick-and-place operations, *Int. J. Robotics Res.* 2(1983) (4), pp. 19-40.

[BLP] R.A. Brooks and T. Lozano Perez, A Subdivision Algorithm in Configuration Space for Findpath with Rotation, AI Memo 684, M.I.T., December 1982.

[Ca] J. Canny, Constructing roadmaps of semi-algebraic sets, preprint, 1986.

[CR] J. Canny and J. Reif, private communication.

[Ch] L.P. Chew, Planning the shortest path for a disc in $O(n^2 \log n)$ time, *Proc. ACM Symp. on Computational Geometry,* 1985, pp. 214-223.

[CS] R. Cole and M. Sharir, Visibility of a polyhedral surface from a point, in preparation.

[Co] G.E. Collins, Quantifier elimination for real closed fields by cylindrical algebraic decomposition, in *Second GI Conference on Automata Theory and Formal Languages,* Lecture Notes in Computer

Science, Vol. 33, pp. 134-183, Springer Verlag, Berlin.

[Da] H. Davenport, A combinatorial problem connected with differential equations, II, *Acta Arithmetica* 17 (1971), pp. 363-372.

[DS] H. Davenport and A. Schinzel, A combinatorial problem connected with differential equations, *Amer. J. Math.* 87 (1965), pp. 684-694.

[FWY] S. Fortune, G. Wilfong and C.K. Yap, Coordinated motion of two robot arms, *Proc. 1986 IEEE International Symposium on Robotics and Automation*, pp. 1216-1223.

[GHLST] L. Guibas, J. Hershberger, D. Leven, M. Sharir, and R.E. Tarjan, Linear time algorithms for shortest path and visibility problems inside simple polygons, *Proc. 2nd ACM Symp. on Computational Geometry*, 1986, pp. 1-13.

[HS] S. Hart and M. Sharir, Nonlinearity of Davenport-Schinzel Sequences and of Generalized Path Compression Schemes, Tech. Report 11/84, The Eskenasy Institute of Computer Sciences, Tel Aviv University, August 1984.

[HJW1] J.E. Hopcroft, D.A. Joseph and S.H. Whitesides, On the Movement of Robot Arms in 2-Dimensional Bounded Regions, *SIAM J. Computing* 14 (1985), pp. 315-333.

[HJW2] J.E. Hopcroft, D.A. Joseph and S.H. Whitesides, Movement Problems for 2-Dimensional Linkages, *SIAM J. Computing* 13 (1984), pp. 610-629.

[HSS1] J.E. Hopcroft, J.T. Schwartz and M. Sharir, On the Complexity of Motion Planning for Multiple Independent Objects; PSPACE Hardness of the 'Warehouseman's Problem', *Int. J. Robotics Research* 3(4) (1984), pp. 76-88.

[HSS2] J.E. Hopcroft, J.T. Schwartz and M. Sharir, Efficient Detection of Intersections Among Spheres, *Int. J. Robotics Research* 2(1983) (4), pp. 77-80.

[HSS3] J.E. Hopcroft, J.T. Schwartz and M. Sharir (Editors) *Planning and Complexity of Robot Motion*, Ablex Pub. Co. (in press).

[HW] J.E. Hopcroft and G. Wilfong, On the Motion of Objects in Contact, Tech. Rept. 84-602, Comp. Science Dept., Cornell University, May 1984.

[IIM] H. Imai, M. Iri and K. Murota, Voronoi diagram in the Laguerre geometry and its applications, *SIAM J. Computing* 14 (1985), pp. 93-105.

[JP] D.A. Joseph and W.H. Plantinga, On the Complexity of Reachability and Motion Planning Questions, *Proc. ACM Symp. on Computational Geometry* 1985, pp. 62-66.

[KLPS] K. Kedem, R. Livne, J. Pach and M. Sharir, On the union of Jordan regions and collision-free translational motion amidst polygonal obstacles, *Discrete and Computational Geometry* 1 (1986), pp. 59-71.

[KS1] K. Kedem and M. Sharir, An Efficient Algorithm for Planning Collision-free Translational Motion of a Convex Polygonal Object in 2-Dimensional Space Amidst Polygonal Obstacles, *Proc. ACM Symp. on Computational Geometry* 1985, pp. 75-80.

[KS2] K. Kedem and M. Sharir, An Efficient Motion-planning Algorithm for a Convex Polygonal Object in Two-dimensional Polygonal Space, in preparation.

[Ke] A. B. Kempe, On a general method of describing plane curves of the n-th degree by linkwork, *Proc. London Math. Soc.* 7 (1876), pp. 213-216.

[LeP] D.T. Lee and F.P. Preparata, Euclidean shortest paths in the presence of rectilinear barriers, *Networks* 14 (1984), pp. 393-410.

[LS1] D. Leven and M. Sharir, An Efficient and Simple Motion Planning Algorithm for a Ladder Moving in Two-Dimensional Space Amidst Polygonal Barriers, *Proc. ACM Symp. on Computational Geometry* 1985, pp. 221-227.

[LS2] D. Leven and M. Sharir, Planning a Purely Translational Motion for a Convex Object in Two-dimensional Space Using Generalized Voronoi Diagrams, Tech. Rept. 34/85, The Eskenasy Institute of Computer Science, Tel Aviv University, June 1985.

[LS3] D. Leven and M. Sharir, On the number of critical free contacts of a convex polygonal object moving in 2-D polygonal space, Tech. Rept. 187, Computer Science Dept., Courant Institute, New York University, October 1985.

[LS4] D. Leven and M. Sharir, Intersection and proximity problems and Voronoi diagrams, to appear in *Advances in Robotics,* Vol. I (J. Schwartz and C. Yap, Eds.), 1986.

[LP] T. Lozano-Perez, Spatial planning: A configuration space approach, *IEEE Trans. Computers* C-32(1983) (2), pp. 108-119.

[LP2] T. Lozano-Perez, A simple motion planning algorithm for general robot manipulators, Tech. Report, AI Lab, M.I.T., 1986.

[LPW] T. Lozano-Perez and M. Wesley, An algorithm for planning collision-free paths among polyhedral obstacles, *Comm. ACM* 22 (1979), pp. 560-570.

[LuS] V.J. Lumelsky and A. Stepanov, Path planning strategies for a traveling automaton in an environment with uncertainty, Tech. Rept. 8504, Center for Systems Science, Yale University, April 1985.

[MY] S. Maddila and C.K. Yap, Moving a polygon around the corner in a corridor, *Proc. 2nd ACM Symp. on Computational Geometry* (1986), pp. 187-192.

[Me] K. Mehlhorn, *Data Structures and Algorithms. III. Multidimensional Searching and Computational Geometry*, Springer Verlag, New York 1984.

[MMP] J. Mitchell, D. Mount and C. Papadimitriou, The discrete geodesic problem, Tech. Rept., Dept. of Operations Research, Stanford University, 1985.

[MP] J. Mitchell and C. Papadimitriou, The weighted region problem, Tech. Rept., Dept. of Operations Research, Stanford University, 1985.

[Mo1] H.P. Moravec, *Robot Rover Visual Navigation*, Ph.D. Dissertation, Stanford University (UMI Research Press, 1981).

[Mo2] D.M. Mount, On finding shortest paths on convex polyhedra, Tech. Rept., Computer Science Department, University of Maryland, October 1984.

[OY] O'Dunlaing, C. and Yap, C., A 'retraction' method for planning the motion of a disc, *J. of Algorithms* 6 (1985), pp. 104-111.

[OSY1] O'Dunlaing, C., Sharir, M. and Yap, C., Generalized Voronoi diagrams for a ladder: I. Topological analysis, Tech. Rept. 139, Computer Science Dept., Courant Institute, New York University, November 1984. (to appear in *Comm. Pure Appl. Math.)*

[OSY2] O'Dunlaing, C., Sharir, M. and Yap, C., Generalized Voronoi Diagrams for a Ladder: II. Efficient Construction of the Diagram, Tech. Rept. 140, Computer Science Dept., Courant Institute, New York University, November 1984. (to appear in *Algorithmica.)*

[OR] J. O'Rourke, Lower bounds on moving a ladder, Tech. Rept. 85/20, Dept. of EECS, The Johns Hopkins University, 1985.

[Pa] C. Papadimitriou, An algorithm for shortest path motion in three dimensions, *Inf. Proc. Letters* 20 (1985), pp. 259-263.

[PSS] R. Pollack, M. Sharir and S. Sifrony, Separating two simple polygons by a sequence of translations, Tech. Rept. 215, Computer Science Dept., Courant Institute, New York University, April 1986.

[Re] J.H. Reif, Complexity of the Mover's Problem and Generalizations, *Proc. 20th IEEE Symp. on Foundations of Computer Science* 1979, pp. 421-427.

[RSh] J.H. Reif and M. Sharir, Motion Planning in the Presence of Moving Obstacles, Tech. Rept. 39/85, The Eskenasy Institute of Computer Science, Tel Aviv University, August 1985.

[RS] J.H. Reif and J.A. Storer, Shortest paths in Euclidean space with polyhedral obstacles, Tech. Rept. CS-85-121, Computer Science Dept., Brandeis University, Waltham, Mass., April 1985.

[dRLW] P.J. deRezende, D.T. Lee and Y.F. Wu, Rectilinear shortest paths with rectangular barriers, *Proc. ACM Symp. on Computational Geometry,* 1985, pp. 204-213.

[SS1] J.T. Schwartz and M. Sharir, On the Piano Movers' Problem: I. The Case of a Two-dimensional Rigid Polygonal Body Moving Amidst Polygonal Barriers, *Comm. Pure Appl. Math.* 36 (1983), pp. 345-398.

[SS2] J.T. Schwartz and M. Sharir, On the "Piano Movers" Problem, General Techniques for Computing Topological Properties of Real Algebraic Manifolds, *Advances in Applied Mathematics* 4 (1983), pp. 298-351.

[SS3] J.T. Schwartz and M. Sharir, On the Piano Movers' Problem: III. Coordinating the Motion of Several Independent Bodies: The Special Case of Circular Bodies Moving Amidst Polygonal Barriers, *Int. J. Robotics Research,* 2 (1983) (3), pp. 46-75.

[SS4] J.T. Schwartz and M. Sharir, On the Piano Movers' Problem: V. The Case of a Rod Moving in Three-dimensional Space Amidst Polyhedral Obstacles, *Comm. Pure Appl. Math.* 37 (1984), pp. 815-848.

[SS5] J.T. Schwartz and M. Sharir, Efficient motion planning algorithms in environments of bounded local complexity, Tech. Rept. 164, Comp. Science Dept., Courant Institute, New York University, June 1985.

[SS6] J.T. Schwartz and M. Sharir, Mathematical Problems and Training in Robotics, *Notices Amer. Math. Soc.* 30 (1983), pp. 478-481.

[Sh1] M. Sharir, Intersection and closest-pair problems for a set of planar discs, *SIAM J. Computing* 14 (1985), pp. 448-468.

[Sh2] M. Sharir, Almost Linear Upper Bounds on the Length of General Davenport-Schinzel Sequences, Tech. Report 29/85, The Eskenasy Institute of Computer Sciences, Tel Aviv University, February 1985.

[Sh3] M. Sharir, On Shortest Paths Amidst Convex Polyhedra, Tech. Rept. 181, Comp. Science Dept., Courant Institute, New York University, September 1985.

[Sh4] M. Sharir, On the two-dimensional Davenport Schinzel problem, Tech. Rept. 193, Comp. Science Dept., Courant Institute, New York University, December 1985.

[Sh5] M. Sharir, Improved lower bounds on the length of Davenport Schinzel sequences, Tech. Rept. 204, Comp. Sci. Dept., Courant Institute, New York University, February 1986.

[SA] M. Sharir and E. Ariel-Sheffi, On the Piano Movers' Problem: IV. Various Decomposable Two-dimensional Motion Planning Problems. *Comm. Pure Appl. Math.* 37 (1984), pp. 479-493.

[SL] M. Sharir and R. Livne, On Minima of Functions, Intersection Patterns of Curves, and Davenport-Schinzel Sequences, *Proc. 26th FOCS Symposium,* 1985, pp. 312-320.

[ShS] M. Sharir and A. Schorr, On shortest paths in polyhedral spaces, *SIAM J. Computing.* 15 (1986), pp. 193-215.

[SiS] S. Sifrony and M. Sharir, An Efficient Motion Planning Algorithm for a Rod Moving in Two-dimensional Polygonal Space, Tech. Rept. 85-40, The Eskenasy Institute of Computer Sciences, Tel Aviv University, August 1985.

[SY] P. Spirakis and C.K. Yap, Strong NP-hardness of moving many discs, *Inf. Proc. Letters* 19 (1984) pp. 55-59.

[SM] K. Sutner and W. Maass, Motion planning among time dependent obstacles, preprint, 1985.

[Sz] E. Szemeredi, On a Problem by Davenport and Schinzel, *Acta Arithmetica* 25 (1974), pp. 213-224.

[TV] R.E. Tarjan and C. Van Wyk, A linear time algorithm for triangulating simple polygons, preprint, October 1985.

[Ud] S. Udupa, Collision Detection and Avoidance in Computer-controlled Manipulators, Ph.D. Dissertation, Calif. Inst. of Technology, Pasadena, Calif. 1977.

[We] Welzl, E., Constructing the visibility graph for n line segments in $O(n^2)$ time, *Inf. Proc. Letters* 20 (1985), pp. 167-172.

[Ya1] C.K. Yap, Coordinating the Motion of Several Discs, Tech. Rept. 105, Computer Science Dept., Courant Institute, New York University, February 1984.

[Ya2] C.K. Yap, An $O(n \log n)$ algorithm for the Voronoi diagram of a set of simple curve segments, Tech. Rept. 161, Computer Science Dept., Courant Institute, New York University, May 1985.

[Ya3] C.K. Yap, Algorithmic motion planning to appear in *Advances in Robotics*, Vol. I (J. Schwartz and C. Yap, Eds.), 1986.

[Ya4] C.K. Yap, How to move a chair through a door, preprint, 1984.

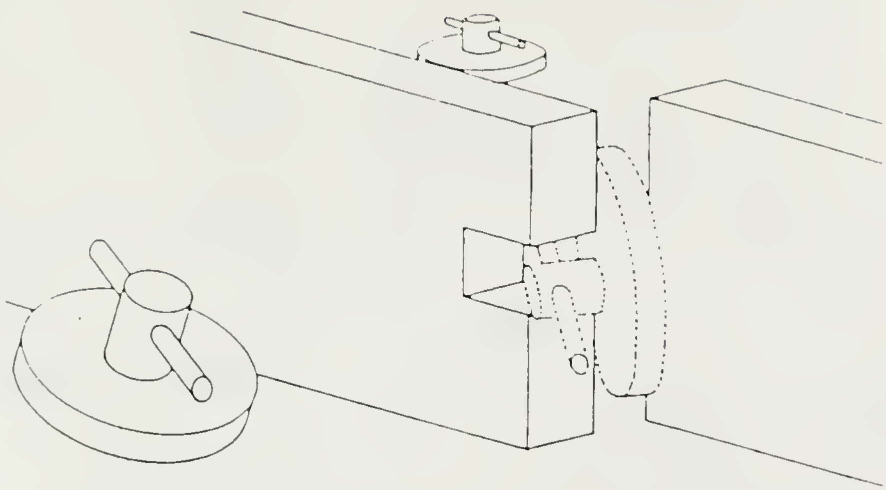


Fig. 1.

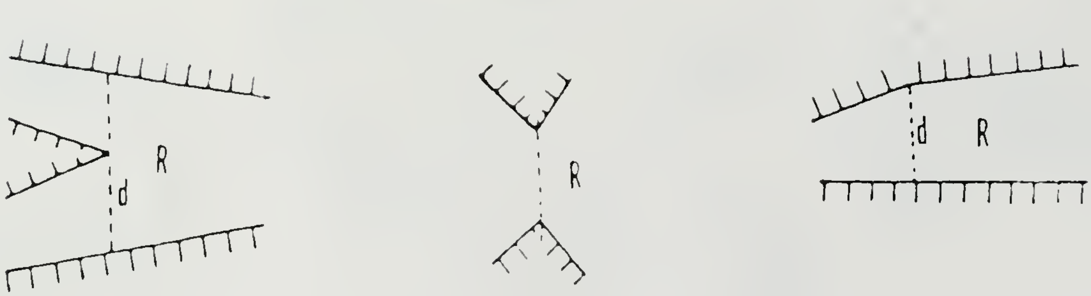


Fig. 2.

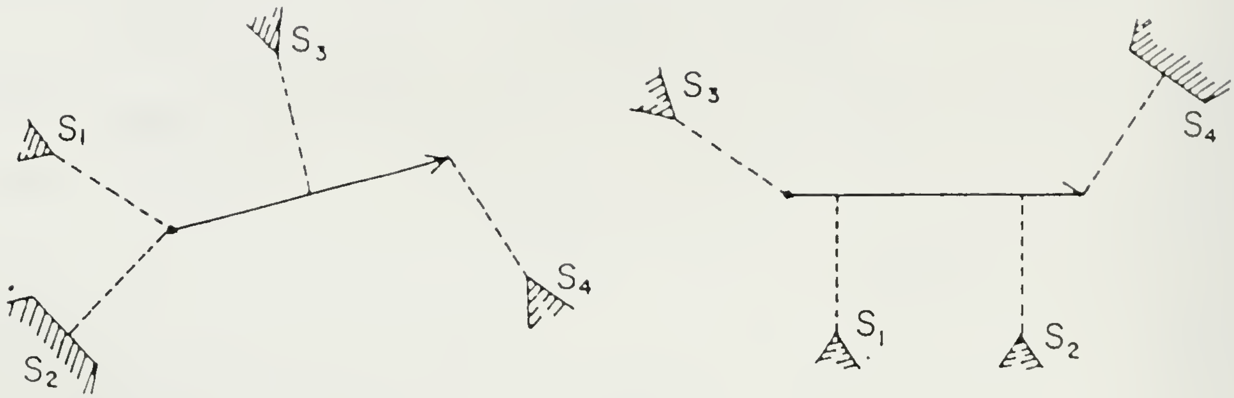


Fig. 3.

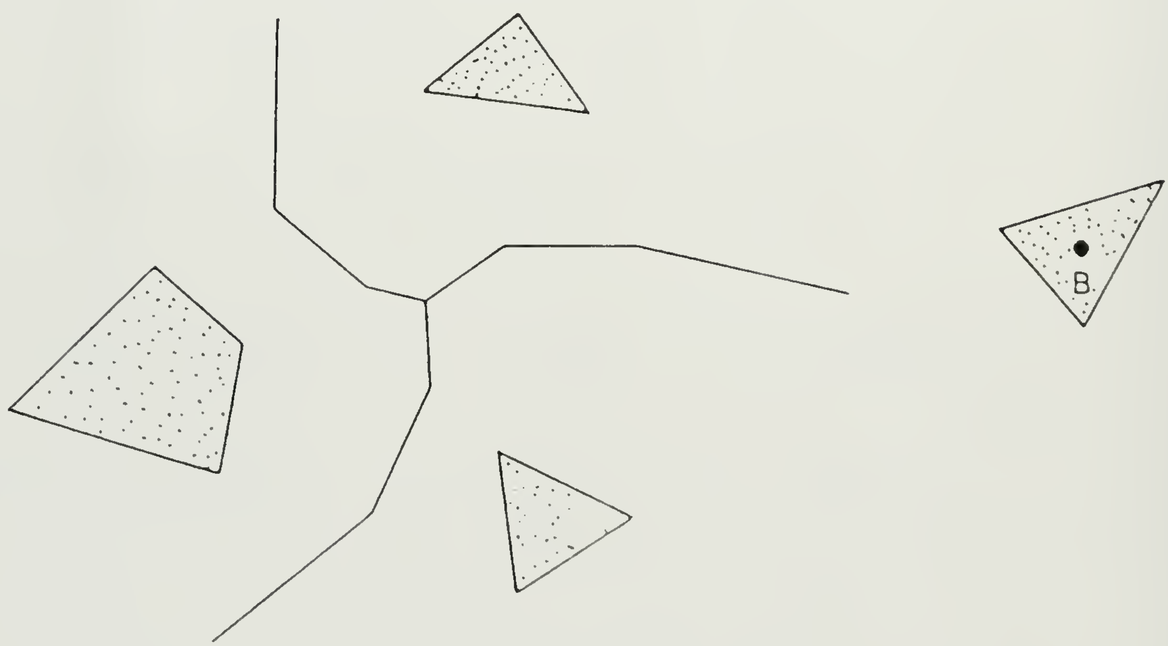


Fig. 4

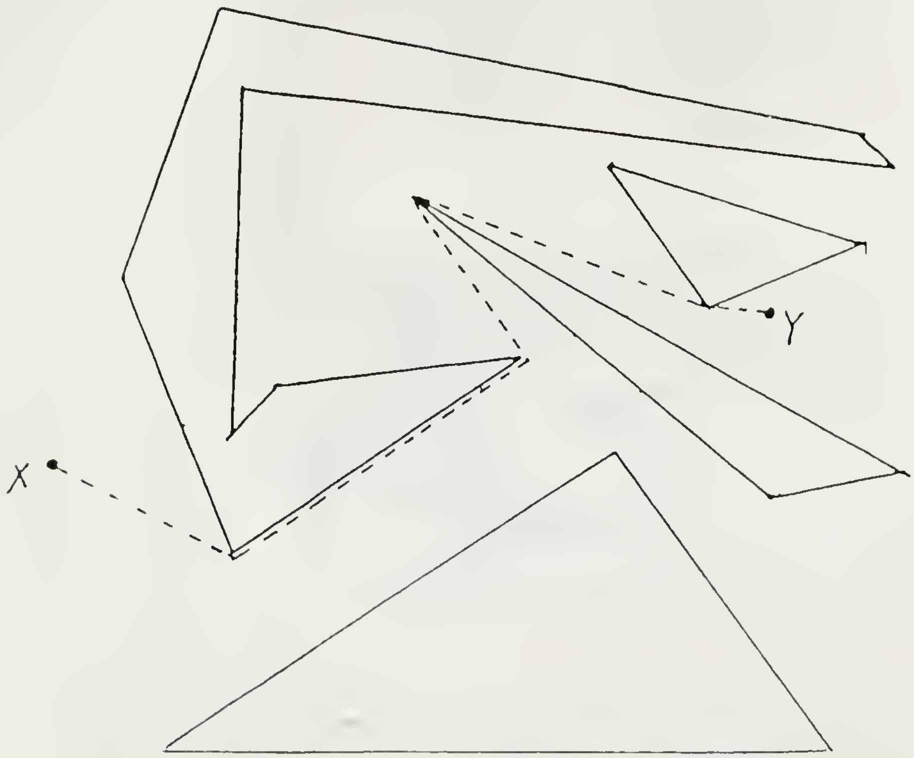


Fig. 5.

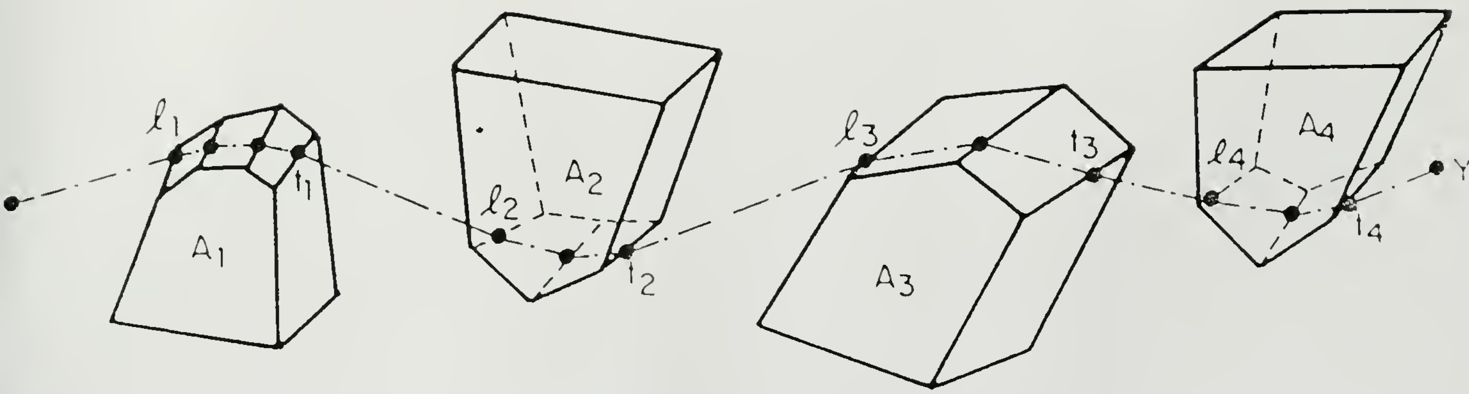


Fig. 6.

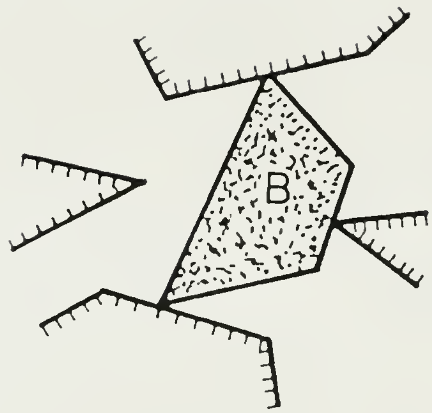


Fig. 7.

NYU COMPSCI TR-241
Schwartz, Jacob T
Motion planning and relate
geometric algorithms in
robotics c.1